

PATENT ABSTRACTS OF JAPAN

(11)Publication number : 07-044487

(43)Date of publication of application : 14.02.1995

(51)Int.Cl.

G06F 13/18

(21)Application number : 05-185129 (71)Applicant : TOSHIBA CORP

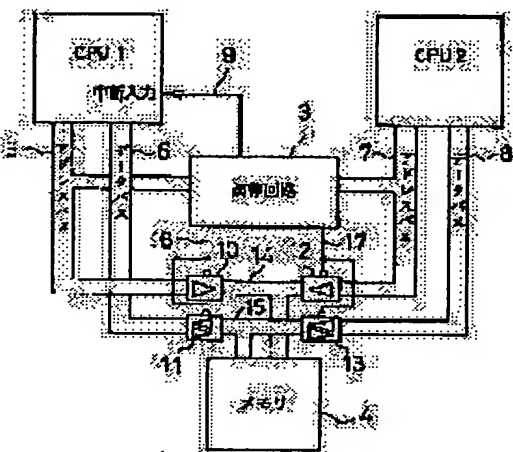
(22)Date of filing : 27.07.1993 (72)Inventor : MATSUDA TETSUJI

(54) MULTI-PROCESSOR SYSTEM

(57)Abstract:

PURPOSE: To provide a multi-processor system in which plural processors are arbitrated to share a resource in common efficiently by a simple configuration additional circuit with respect to the multi-processor system using plural independent processors to make high speed processing.

CONSTITUTION: The system is configured by using plural independent processor CPU_i (i=1, 2), a common resource 4 shared in common by the plural processors CPU_i, and an arbitration means 3 applying allocation control so as to allow one CPU_i in the plural processors to access the common resource 4, and the arbitration means 3 outputs an interrupt signal 9 commanding the interruption of the access of the processor CPU_i to the common resource 4.



LEGAL STATUS

[Date of request for examination]

[Date of sending the examiner's decision of rejection]

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]

[Date of final disposal for application]

[Patent number]

[Date of registration]

[Number of appeal against examiner's decision of rejection]

(19)日本国特許庁 (J P)

(12) 公 開 特 許 公 報 (A)

(11)特許出願公開番号

特開平7-44487

(43)公開日 平成7年(1995)2月14日

(51)Int.Cl.⁶

G 0 6 F 13/18

識別記号

5 1 0 B 9366-5B

庁内整理番号

F I

技術表示箇所

審査請求 未請求 請求項の数4 O L (全 9 頁)

(21)出願番号 特願平5-185129
(22)出願日 平成5年(1993)7月27日

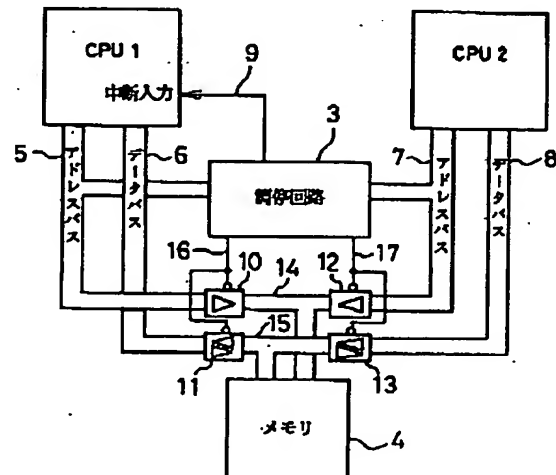
(71)出願人 000003078
株式会社東芝
神奈川県川崎市幸区堀川町72番地
(72)発明者 松田 哲治
神奈川県川崎市幸区堀川町580番1号 株
式会社東芝半導体システム技術センター内
(74)代理人 弁理士 三好 保男 (外3名)

(54)【発明の名称】 マルチプロセッサシステム

(57)【要約】

【目的】 複数の互いに独立したプロセッサを用いて高速処理を行うマルチプロセッサシステムに関し、簡単な構成の付加回路によって、複数のプロセッサが効率よく資源を共有するように調停できるマルチプロセッサシステムを提供することを目的とする。

【構成】 複数の互いに独立したプロセッサCPU1 (1=1, 2) と、複数のプロセッサCPU1が共有する共有資源4と、複数のプロセッサの内1つCPU1を共有資源4に対してアクセス可能となるよう割当制御する調停手段3とを有して構成し、調停手段3は、少なくとも1つのプロセッサCPU1に対して、該プロセッサCPU1の共有資源4に対するアクセスを中断するよう指示する中断信号9を出力することである。



【特許請求の範囲】

【請求項1】 複数の互いに独立したプロセッサと、前記複数のプロセッサが共有する共有資源と、前記複数のプロセッサの内1つを前記共有資源に対してアクセス可能となるよう割当制御する調停手段とを有し、前記調停手段は、前記少なくとも1つのプロセッサに対して、該プロセッサの前記共有資源に対するアクセスを中断するよう指示する中断信号を出力することを特徴とするマルチプロセッサシステム。

【請求項2】 前記調停手段は、前記割当制御を前記プロセッサの持つ優先順位に従って行うことを特徴とする請求項1に記載のマルチプロセッサシステム。

【請求項3】 前記プロセッサは、前記中断信号によって中断された前記共有資源に対するアクセスを、中断された状態から再実行可能であることを特徴とする請求項1または2に記載のマルチプロセッサシステム。

【請求項4】 前記プロセッサは、中央処理装置（CPU）、またはバスマスタであることを特徴とする請求項1、2、または3に記載のマルチプロセッサシステム。

【発明の詳細な説明】

【0001】

【発明の技術分野】 本発明は、複数の互いに独立したプロセッサを用いて高速処理を行うマルチプロセッサシステムに関し、特に、簡単な構成の付加回路によって、複数のプロセッサが効率よく資源を共有するように調停可能としたマルチプロセッサシステムに関する。

【0002】 近年、中央処理装置（CPU）等のプロセッサの動作周波数は年々高速化されてきている。一方、周辺装置のアクセス速度についても向上しているものの、中央処理装置の速度に追いついて来ないのが現状である。

【0003】 このように、中央処理装置内部の処理速度と外部周辺装置に対するアクセス速度との速度比は拡大の傾向にあり、共有資源を備えるマルチプロセッサシステムにおいて、優先度の低いプロセッサの共有資源に対するアクセスが終了するまで、優先度の高いプロセッサが待つ待ち時間が、システム全体のスループットにより大きな影響を及ぼすことが予想され、システム全体のスループットを低下させない資源割当の制御方法が望まれている。

【0004】

【従来の技術】 従来のマルチプロセッサシステムの構成図を図6に示す。

【0005】 同図に示すマルチプロセッサシステムは、互いに独立した2つのプロセッサCPU1及びCPU2によって構成され、2つのプロセッサCPU1及びCPU2はメモリ4を共有している。また本従来例の構成では、メモリ4に対するアクセスの優先順位は、プロセッサCPU2の方が高いものとする。

【0006】 また、調停回路3は、プロセッサCPU1及びCPU2が共有資源（メモリ4）を同時にアクセスしようとした場合、プロセッサCPU2のアクセス要求を選択し、プロセッサCPU1を待機させるように制御する。プロセッサCPU1に関して、アドレスバス5が調停回路3、1方向バスドライバ10、及び共有アドレスバス14を介してメモリ4に、データバス6が双方向バスドライバ11及び共有データバス15を介してメモリ4に、それぞれ接続されている。

10 【0007】 同様にして、プロセッサCPU2に関して、アドレスバス7が調停回路3、1方向バスドライバ12、及び共有アドレスバス14を介してメモリ4に接続され、データバス8が双方向バスドライバ13及び共有データバス15を介してメモリ4に接続されている。

【0008】 プロセッサCPU1が共有資源であるメモリ4をアクセス中に、優先度の高いプロセッサCPU2のメモリ4に対するアクセス要求があった場合のタイミングチャートを図7に示す。

【0009】 本従来例では、プロセッサCPU1がメモリ4に対するアクセス権を受けて、バスサイクルをT1のタイミングから実行中である時に、それよりも優先度の高いプロセッサCPU2が、T2のタイミングからメモリ4をアクセスしようとしても、優先度の低いプロセッサCPU1が実行中のバスサイクルを終了するT3のタイミングからでしか、次のアクセス要求を受け付けられないようになっていた。

【0010】 また、バスサイクルを中断すると、中断されたバスサイクル中のデータが失われるため、いずれバスサイクルを再開する際に、中断したバスサイクルを再実行する必要があった。しかしながら、再実行のための機能を付加すると調停回路3等のハードウェア量が大きくなってしまうため、バスサイクル実行途中での中断は行われていなかった。尚、バリエーション等で正しいデータが得られなかった場合に、同じバスサイクルを再実行するリトライ動作を行うための入力を持つプロセッサは、従来技術でも開発されている。

【0011】

【発明が解決しようとする課題】 以上のように、従来のマルチプロセッサシステムでは、実行中のバスサイクルを中断できないシステム構成のため、アクセス権を獲得する際のオーバーヘッド時間が大きくなってしまい、システム全体としての処理効率が落ちるという問題があった。

【0012】 本発明は、上記問題点を解決するもので、簡単な構成の付加回路によって、複数のプロセッサが効率よく資源を共有するように調停可能としたマルチプロセッサシステムを提供することを目的とする。

【0013】

【課題を解決するための手段】 前記課題を解決するため
50 に、本発明のマルチプロセッサシステムの第1の特徴

3

は、図1に示す如く、複数の互いに独立したプロセッサCPU1(1=1, 2)と、前記複数のプロセッサCPU1が共有する共有資源4と、前記複数のプロセッサの内1つCPU1を前記共有資源4に対してアクセス可能となるよう割当制御する調停手段3とを具備し、前記調停手段3は、前記少なくとも1つのプロセッサCPU1に対して、該プロセッサCPU1の前記共有資源4に対するアクセスを中断するよう指示する中断信号9を出力することである。

【0014】また、本発明の第2の特徴は、請求項1に記載のマルチプロセッサシステムにおいて、前記調停手段3は、前記割当制御を前記プロセッサCPU1の持つ優先順位に従って行うことである。

【0015】また、本発明の第3の特徴は、請求項1または2に記載のマルチプロセッサシステムにおいて、前記プロセッサCPU1は、前記中断信号9によって中断された前記共有資源4に対するアクセスを、中断された状態から再実行可能であることである。

【0016】更に、本発明の第4の特徴は、請求項1、2、または3に記載のマルチプロセッサシステムにおいて、前記プロセッサCPU1は、中央処理装置(CPU)、またはバスマスタであることである。

【0017】

【作用】本発明の第1、第2、第3、及び第4の特徴のマルチプロセッサシステムでは、図1に示す如く、複数の互いに独立したプロセッサCPU1(1=1, 2)が特定の資源4を共有するマルチプロセッサシステムの資源割当制御において、優先度の低いプロセッサCPU1が実行中のバスサイクルよりも、優先度の高いプロセッサCPUj(j≠1)からのバスアクセス要求があった場合に、現在実行中の優先度の低いプロセッサCPU1のバスサイクルを中断し、優先度の高いプロセッサCPUjのバスサイクルを優先的に行う調停方式を採用している。

【0018】調停手段3が行う資源割当制御は、少なくとも1つのプロセッサCPU1に対して、該プロセッサCPU1の共有資源4に対するバスアクセスを中断するよう指示する中断信号9を出力して、該プロセッサCPU1のバスサイクルを実行途中であっても強制的に中断させる。

【0019】以上のように、簡単な構成の付加回路により、優先度の高いプロセッサのアクセス待ちのオーバーヘッド時間を減少でき、マルチプロセッサシステム全体のスループットの低下を抑ええることができ、複数のプロセッサが効率よく資源を共有するように調停可能なマルチプロセッサシステムを実現できる。

【0020】

【実施例】以下、本発明に係る実施例を図面に基いて説明する。

【0021】図1に本発明の第1の実施例に係るマルチ

4

プロセッサシステムのシステム構成図を示す。

【0022】本実施例のマルチプロセッサシステムは、互いに独立した2つのプロセッサCPU1及びCPU2によって、メモリ4を共有している。プロセッサCPU1には、バスサイクルを実行途中で強制的に中断する入力端子として、バリティエラー等で繰り返し同じバスサイクルを実行し続けるための入力(以下、「中断入力という」)端子を備えている。

【0023】また本実施例では、メモリ4に対するアクセスの優先順位は、プロセッサCPU1よりもプロセッサCPU2の方が高いものとする。プロセッサCPU1に関しては、アドレスバス5が、調停回路3及び1方向バスドライバ10と共有アドレスバス14を介してメモリ4に、データバス6が、双方向バスドライバ11と共有データバス15を介してメモリ4にそれぞれ接続されている。

【0024】同様にして、プロセッサCPU2に関しては、アドレスバス7が調停回路3及び1方向バスドライバ12と共有アドレスバス14を介してメモリ4に接続され、データバス8が双方向バスドライバ13と共有データバス15を介してメモリ4に接続されている。更に、プロセッサCPU1の中断入力には、調停回路3からの中断信号9が接続されている。

【0025】本実施例の調停回路3の回路例を図2に示す。プロセッサCPU1及びCPU2のアドレス信号は、それぞれデコーダ105及び107によりデコードされる。デコーダ105及び107はいわゆるアドレスデコーダであり、プロセッサCPU1及びCPU2がそれぞれ共有資源としているメモリ4をアクセスしようとした場合、デコーダ105及び107からの出力信号101及び17がイネーブルになる。

【0026】論理ゲート109は、プロセッサCPU1がメモリ4に対してアクセスしており、プロセッサCPU2がメモリ4に対してアクセスをしていない時に信号線16をイネーブルにする。論理ゲート103は、プロセッサCPU1及びCPU2が同時にメモリ4に対してアクセスをしている間、中断信号9をイネーブルにする中断信号出力回路である。

【0027】プロセッサCPU1がメモリ4にアクセス中に、プロセッサCPU2のメモリ4へのアクセスが始まった場合について、図3に示すタイミングチャートを参照して説明する。

【0028】まず、プロセッサCPU1のみが共有資源であるメモリ4に対して、T1のタイミングからアドレスバスにアドレスAを出力しバスサイクルの実行を開始する。この時、1方向バスドライバ10がアクティブになり共有アドレスバス14にアドレスAが出力される。次にこのバスサイクル中のT2のタイミングで、プロセッサCPU2がメモリ4をアクセスするためアドレスバス7にアドレスBを出力する。これを受けて調停回路3

は、中断信号9を出力することによってプロセッサCPU1が実行中のバスサイクルを中断する。

【0029】従って、1方向バスドライバ10がインアクティブになり、1方向バスドライバ12がアクティブになって、アドレスBが共有アドレスバス14に出力される。つまりプロセッサCPU2はメモリ4に対するアクセス権を得て、T2のタイミングからアクセスを始めることになる。

【0030】プロセッサCPU2がメモリ4をアクセス中は、調停装置3からプロセッサCPU1に中断信号9が出力され続ける。本実施例では、プロセッサCPU2はT2'のタイミングからデータ転送を行っている。プロセッサCPU2がメモリ4のアクセスをT3のタイミングで終了すると、調停回路3は中断信号9をディセーブルにする。

【0031】プロセッサCPU1は、中断信号9がディセーブルになると、中断されていたバスサイクルをT3のタイミングよりもう一度最初から実行し直す。図3の例では、プロセッサCPU1はT3'のタイミングからデータ転送を行っている。また、本実施例では、プロセッサCPU1はバスサイクルを中断している間、停止した待ち状態に入るが、そのバスサイクルを中断している間に、例えば、ローカルメモリ（図示せず）への命令フェッチ等を行いながら待つようにしても良い。尚、ここでいうプロセッサCPU1及びCPU2は、キャッシュメモリ等を内蔵する広義のプロセッサを指す。

【0032】次に、第2の実施例として、n個のプロセッサCPU1～CPU_nを用いて構成した場合のマルチプロセッサシステムのシステム構成図を図4に示す。また、第2の実施例の調停回路の回路図を図5に示す。

【0033】図4において、本実施例のマルチプロセッサシステムは、複数の互いに独立したプロセッサCPU1（i=1～n）と、複数のプロセッサCPU1が共有するメモリ4と、プロセッサCPU1のアドレスバス5-1及び共有アドレスバス14間のアドレスの方向制御を行う1方向バスドライバ10-1と、プロセッサCPU1のデータバス6-1及び共有データバス15間のデータの方向制御を行う双方向バスドライバ11-1と、複数のプロセッサの内1つCPU1をメモリ4に対してアクセス可能となるよう、プロセッサCPU_j（j=1～n-1）に中断信号9-jを、また1方向バスドライバ10-1及び双方向バスドライバ11-1にインエーブル/ディセーブル信号16-1をそれぞれ発して割当制御する調停回路3nとから構成されている。

【0034】また、調停回路3nの構成は図5に示す如く、プロセッサCPU1に対するデコーダ105-1と、デコーダ105-1からの出力信号線101-1の論理演算によりインエーブル/ディセーブル信号16-1を生成する論理ゲート109-1と、デコーダ105-1からの出力信号線101-1の論理演算により中断信

号9-jを生成する論理ゲート103-1及び104-1とから構成されている。

【0035】本実施例のマルチプロセッサシステムは、2個のプロセッサCPU1及びCPU2で構成される第1の実施例を、n個のプロセッサCPU1～CPU_nで構成されるように拡張したものであり、個別のプロセッサCPU1の動作は、第1の実施例と同様である。

【0036】第1及び第2の実施例では、バリティエラー用のリトライ動作を行うために用意された端子を中断信号9の入力として用いたが、本発明の目的から言って、この端子は特にリトライ動作である必要はない。

【0037】以上のように、第1及び第2の実施例のマルチプロセッサシステムによれば、複数のプロセッサCPU1（i=1～n）の共有に係るメモリ4をアクセスしようとする時、既に、あるプロセッサCPU1がバスアクセス中だった場合でも、該プロセッサCPU1に対して中断信号9-1を発することにより、そのバスサイクルを強制的に中断させることができる。

【0038】その結果、従来例のタイミング（図7）と本発明の第1の実施例のタイミング（図3）を比較すると、従来例に比べて第1の実施例の方が、優先度の高いプロセッサCPU2のアクセスを3システムクロック早く始めることができる。以上の理由から、上述した実施例は、優先度の高いプロセッサのアクセス待ちのオーバーヘッド時間を減少でき、マルチプロセッサシステム全体の効率を低下させることがない。

【0039】

【発明の効果】以上のように本発明によれば、複数の互いに独立したプロセッサが特定の資源を共有するマルチプロセッサシステムの資源割当制御において、優先度の低いプロセッサが実行中のバスサイクルよりも、優先度の高いプロセッサからのバスアクセス要求があった場合に、調停手段から中断信号を発して、現在実行中の優先度の低いプロセッサのバスサイクルを強制的に中断し、優先度の高いプロセッサのバスサイクルを優先的に行う調停方式を採用したので、簡単な構成の付加回路により、優先度の高いプロセッサのアクセス待ちのオーバーヘッド時間を減少させて、マルチプロセッサシステム全体のスループットの低下を抑ええることができ、複数のプロセッサが効率よく資源を共有するように調停可能なマルチプロセッサシステムを提供することができる。

【図面の簡単な説明】

【図1】本発明の第1の実施例に係るマルチプロセッサシステムのシステム構成図である。

【図2】第1の実施例における調停回路の内部構成例を示す回路構成図である。

【図3】第1の実施例の動作を説明するタイミングチャートである。

【図4】本発明の第2の実施例に係るマルチプロセッサシステムのシステム構成図である。

【図5】第2の実施例における調停回路の回路構成図である。

【図6】従来のマルチプロセッサシステムのシステム構成図である。

【図7】従来例の動作を説明するタイミングチャートである。

【符号の説明】

CPU1, CPU2 プロセッサ

3 調停回路

4 共有メモリ

5 CPU1のアドレスバス

6 CPU1のデータバス

7 CPU2のアドレスバス

8 CPU2のデータバス

9 中断信号(線)

10 1方向バスドライバ

11 双方向バスドライバ

12 1方向バスドライバ

13 双方向バスドライバ

14 共有アドレスバス

15 共有データバス

16 CPU1側のバスドライバのイネーブル/ディセーブル信号線

17 CPU2側のバスドライバのイネーブル/ディセーブル信号線

105, 107 デコーダ

101 デコーダ105からの出力信号線

103, 109 論理ゲート

CPU1~CPU_n プロセッサ

3_n 調停回路

5-*i* (*i*=1~*n*) CPU1のアドレスバス

6-*i* (*i*=1~*n*) CPU1のデータバス

10 9-*i* (*i*=1~*n*-1) 第1中断信号(線)

10-*i* (*i*=1~*n*) CPU1に対する1方向バスドライバ

11-*i* (*i*=1~*n*) CPU1に対する双方向バスドライバ

16-*i* (*i*=1~*n*) 第1バスドライバのイネーブル/ディセーブル信号線

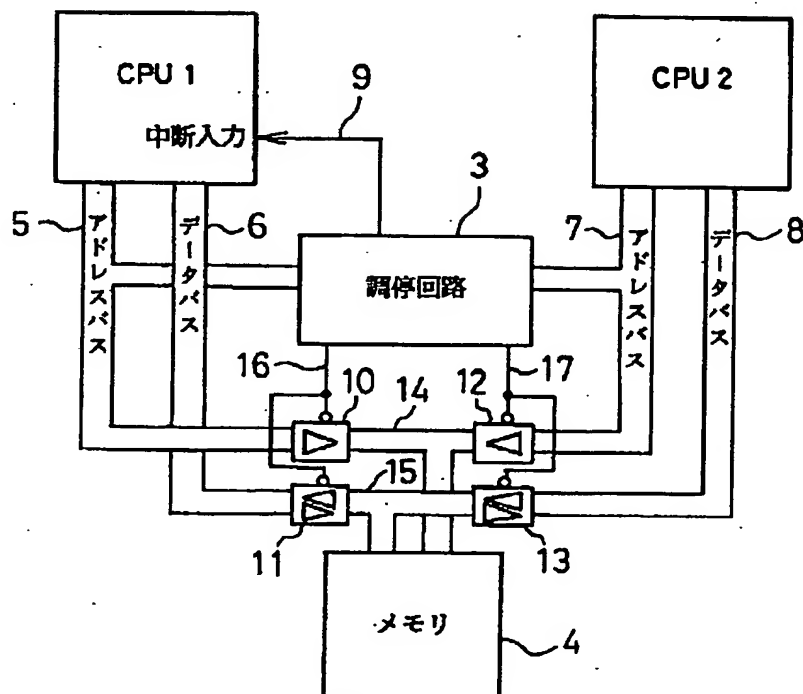
105-*i* (*i*=1~*n*) CPU1に対するデコーダ

101-*i* (*i*=1~*n*) デコーダ105-*i*からの出力信号線

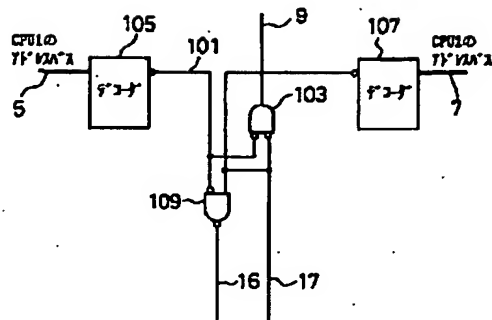
20 103-*i* (*i*=1~*n*-1), 104-*i* (*i*=1~*n*-2) 論理ゲート

109-*i* (*i*=1~*n*-1) 論理ゲート

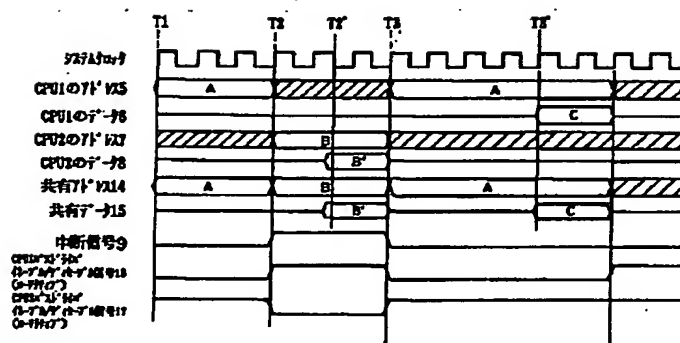
【図1】



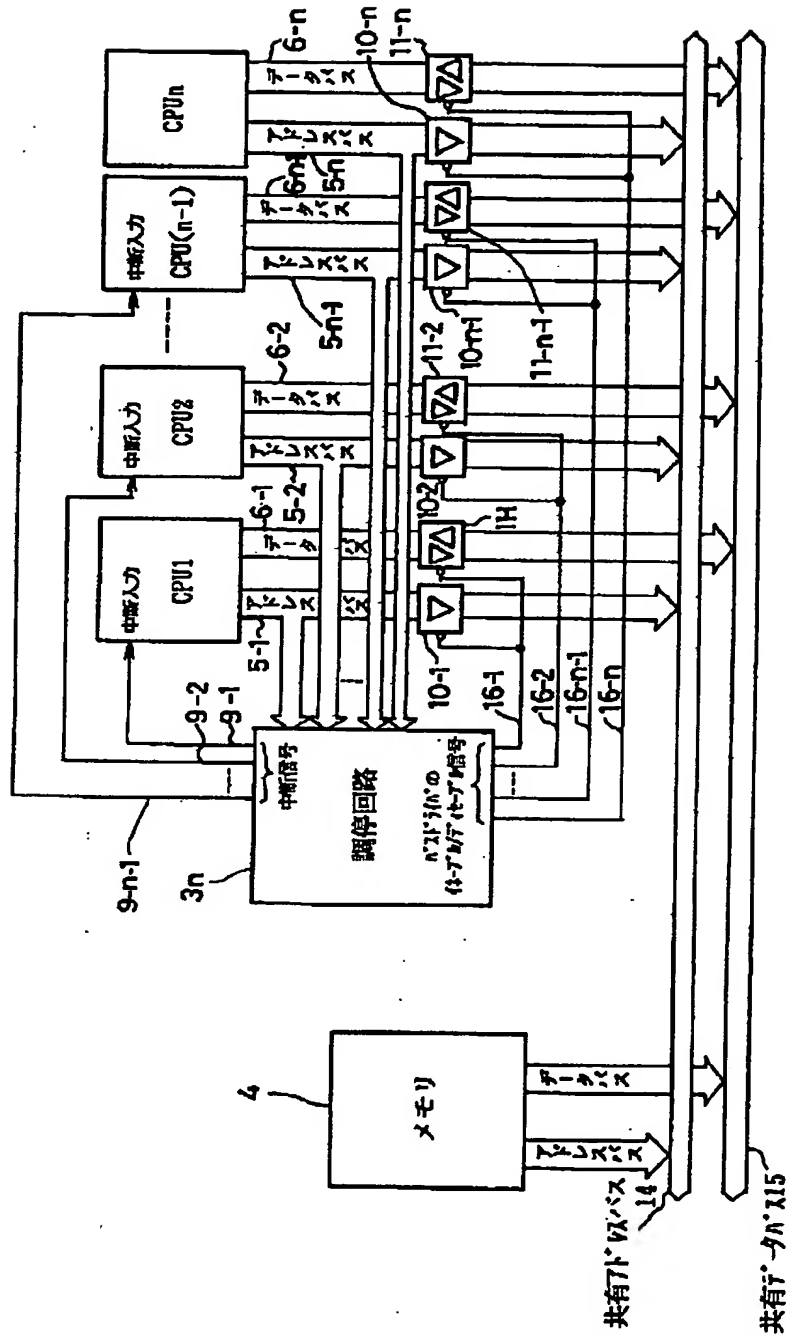
【図2】



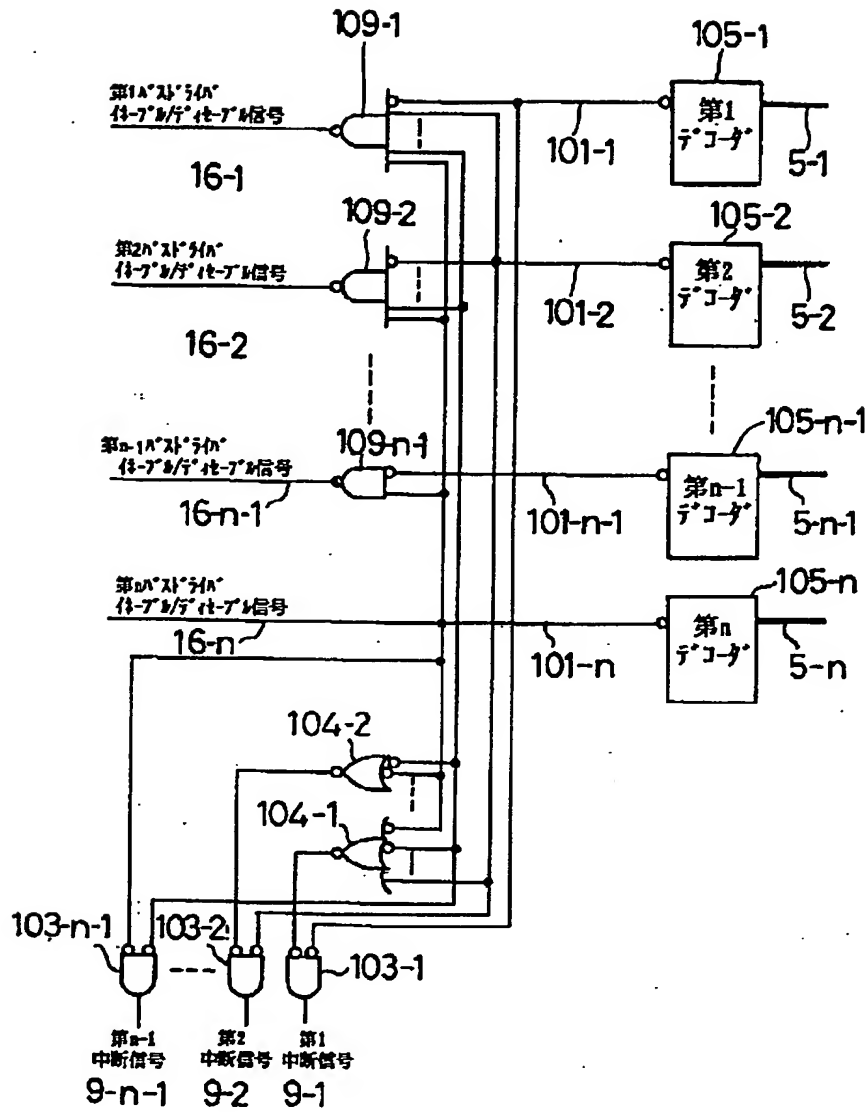
【図3】



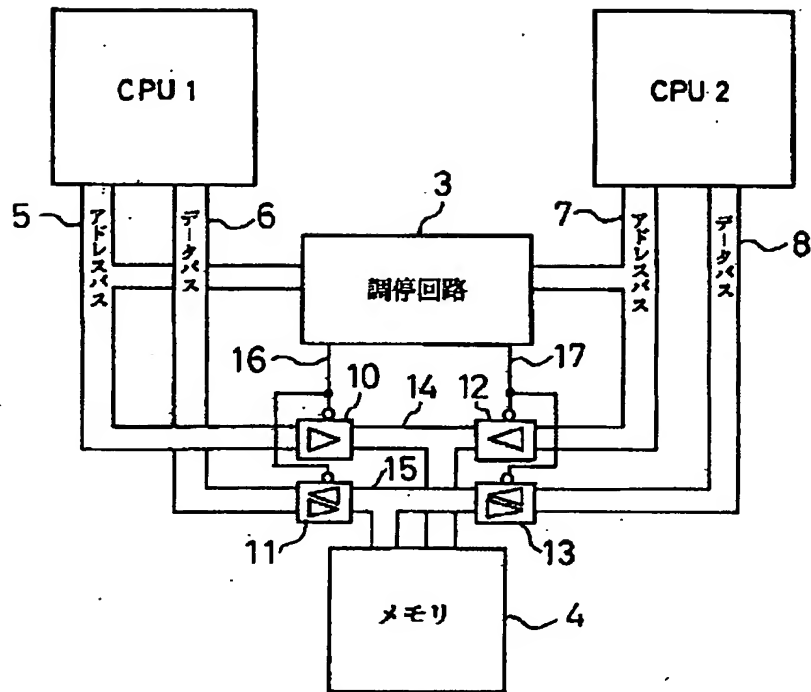
【図4】



【図5】



【図6】



【図7】

